**Lecture 1: Working with Objects and JSON**

**1. Understanding Objects in JavaScript**

In JavaScript, an **object** is a collection of **properties**.  
Each property has a **key** (name) and a **value**.

**a) Object Literals**

We create objects using {} curly braces.  
Example:

const user = {

name: 'John',

age: 25

};

**Explanation:**

* name and age are **keys** (properties).
* 'John' and 25 are their **values**.

**b) Accessing Properties**

There are **two ways**:

**Dot Notation: objectName.propertyName**

console.log(user.name); // Output: John

**Bracket Notation objectName["propertyName"]**

console.log(user['age']); // Output: 25

**When to use bracket notation?**  
When the property name is **dynamic** (comes from a variable).

Example:

const key = 'name';

console.log(user[key]); // Output: John

**c) Nested Objects**

Sometimes, a property itself can be another object (complex data).

Example:

const person = {

name: 'Sara',

address: {

city: 'Lahore',

zip: '54000'

}

};

// Access nested properties

console.log(person.address.city); // Output: Lahore

console.log(person['address']['zip']); // Output: 54000

**2. Object Methods**

Functions that belong to an object are called **methods**.

**a) Defining Methods**

Example:

const user = {

name: 'John',

greet: function() {

return 'Hello ' + this.name;

}

};

console.log(user.greet()); // Output: Hello John

**b) Using the this Keyword**

* this refers to the **current object** inside methods.

console.log(this); // In browser console, it points to window object.

const car = {

brand: 'Toyota',

displayBrand: function() {

console.log(this.brand);

}

};

car.displayBrand(); // Output: Toyota

**Important:**  
Inside methods, this refers to the **object** calling the method.

**3. Introduction to JSON (JavaScript Object Notation)**

JSON is a format to store and exchange **data**. It is **text-based** and looks like JavaScript objects, but:

* Keys and string values must be in **double quotes (" ")**.
* JSON is **used in APIs, data transfer, storage**.

Example of JSON:

{

"name": "John",

"age": 25

}

**a) Converting Objects to JSON**

Use JSON.stringify() ➡️ Converts JavaScript object ➡️ JSON string.

Example:

const user = { name: 'John', age: 25 };

const jsonString = JSON.stringify(user);

console.log(jsonString);

// Output: {"name":"John","age":25}

**b) Parsing JSON Data**

Use JSON.parse() ➡️ Converts JSON string ➡️ JavaScript object.

Example:

const jsonData = '{"name":"John","age":25}';

const userObject = JSON.parse(jsonData);

console.log(userObject.name); // Output: John

console.log(userObject.age); // Output: 25

**c) Use Cases of JSON**

**APIs:** When your app talks to a server, the server usually sends/receives data in JSON.  
**Data Exchange:** Easy way to transfer data between frontend and backend.  
**Storage:** You can store JSON in files, databases, browsers, etc.

**4. Practical Activity: Fetch and Display JSON Data from a Local File**